



**COMPUTER
CENTRE**

BULLETIN

*Volume 4, Number 2.
1st February, 1971.*

*Editor:
S. J. Barry.*

USE OF THE COMPUTER

The Statement on Conditions of Use and Liability contained in January's Bulletin may have come as a surprise to some clients. However, it should be realised that these conditions are not really new, but are a formal statement of the way the Computer Centre operates, and do not really differ much from previous practice. Because of the very significant changes over the last year and the increasing amount of work being undertaken, there has been a need to formalize our arrangements for using the computer.

Here, as in many other enterprises, the key to effective use of the facilities lies in cooperation, and it would indeed be unfortunate if the statement of Conditions of Use inhibited this. The Centre is particularly anxious to secure the cooperation of users in reporting software errors. Although all software is tested before being released it is (regrettably) true that errors do exist, and prompt reporting of them will enable us to take remedial action and thereby improve its quality.

It must be appreciated that complete reliability cannot be achieved by this (or any other) computer system and any use must be based on this premise. The Computer Centre accepts work on the assumption that the user has planned for the possible occurrence of errors. It will run the user's job and will accept responsibility for rerun or credit of a run that was spoiled by a machine, software, or operational error. The user must however ensure that a faulty run can be identified and that the fault will not have repercussions beyond that run. The Computer Centre will not accept liability other than for the particular run that was proved to be in error.

The user must also bear in mind that the situation is not static, particularly where software is concerned. The quantity and complexity of system software has become such that it is impossible to evaluate completely the effect of even simple changes. Changes to software come from many different sources. In addition to those occasioned by the Centre's overall development program, some changes are made to correct reported errors. Material is also received from Digital Equipment and other sources, ranging from complete new releases of software to minor patches. All these changes are tested before being incorporated into the working system. For example, the test deck for the FORTRAN system now amounts to some 2000 statements! Despite all testing, it seems that a few errors remain (or are introduced) with each system change and it is in the location of these that user's cooperation is sought. As errors are detected, they are reported through the Weekly Newsletter and corrective action is initiated. Such action may require that the problem be referred to Digital Equipment, in which case some months can elapse before the error is corrected.

If the Centre can rely on user's support in the reporting of errors and programming around them until they are corrected, then its task is made much easier. The only alternative is to withhold the software until further testing and correction has been completed. New software received is generally a substantial improvement on the older version currently in use and it is desirable to release it for general use as soon as basic testing has been completed, rather than to withhold it pending extended testing, which will certainly take time and may not in any case detect the errors that a subsequent user's program will reveal.

It is hoped that the above comments indicate the necessity for adopting a methodical approach to the use of the computer. It is important for each user to proceed in stages and to ensure that it is possible to revert to a previous stage if there is any possibility that a user change or a system error is creating anomalous results. Test data and results should be available to verify the correct working of a program and, if possible, consistency checks should be built into the program. A qualitative survey of the problems discovered during consultation last year would indicate that a substantial number of computer users are not sufficiently careful in their use of the computer. For the benefit of users, the following remarks are offered:

- (a) If a problem appears, find its cause and correct it. There is very little chance that it will disappear as a consequence of development of the program in other areas.
- (b) Make sure the program has many comments to explain what it is doing. If the user himself cannot understand why he is getting peculiar results, there is little chance that someone else will.
- (c) It is fallacious to expect that a small change to a program will have an insignificant effect on the results. When trying to evaluate the effect of a particular program change, make sure that other seemingly irrelevant changes elsewhere are not clouding the issue.
- (d) Give some thought to the possibility that numerical underflow or overflow has occurred. The present FORTRAN system allows these to happen and may cause some very odd results.

REMOTE TERMINALS

The first sixteen remote terminals were released by the Centre during January and have been running fairly successfully during their first month of operation. During this period the use of terminals was free of charge to give the opportunity of becoming familiar with this new mode of operation. As from Tuesday 2 February 1971, normal charges will be levied for remote terminal work. On this date, passwords will also revert to the one nominated on the Authorization Form. Changes made to passwords during January will be ignored.

The seminars on the remote terminal system have been fairly well attended. Any users who missed the lectures have an opportunity to attend either of the two seminars planned for February. The seminar for the 9 February will be held from 10 a.m. to 1 p.m. in Lecture Room B18 of the Engineering Building. The venue for the seminar on the 23 February has been changed and users will be notified through the Newsletter.

A number of Manuals have been produced to assist users with the work. Unfortunately, due to unavoidable delays these manuals are not all presently available. The University Bookshop currently holds stocks of the FORTRAN and EDITOR manuals. Other manuals to be released shortly include the System User's guide and AID. Their release will be notified in the Newsletter and later Bulletins. A list of all the commands available under the new system is obtainable on request from the Computer Centre.

COMPUTER OPERATIONS

From Monday 18 January computer operations at the Centre were extended to include a second shift. The proposed schedule at the time this issue went to print was as follows:

Timesharing	1000-1200, 1400-1600
Batch	1000-1200, 1400-2300

Although major development testing is done during week ends, the period 1200-1400 is usually reserved for short interval tests and checks. On any day that this reserved time is not required it will be made available for both Timesharing and Batch operation. Users will be advised of the availability of this time via the schedule board and messages through the terminals.

DATA FILES

1. GENERAL

All information within the PDP-10 system is kept in the form of files.

These files can be classified into two very broad categories. Many files contain programs in one form or another and as such are files which can be placed into execution by means of some 'processor program'. These comprise the first category.

The second category consists of files which are simply used for holding information to be read or written by a program. The information in these files is of such a nature that the file could not be placed into execution. These are commonly called data files.

2. DATA FILENAMES

The general form of a filename is

name/processor program name

where the name is the identification given to the file by the user and the processor program is the name of the routine necessary to place the program file into execution.

As a data file cannot be placed into execution a data filename has no 'processor program' part. It is simply an identification of the form

name

examples:

- (i) DATA
- (ii) INPUT
- (iii) OUT
- (iv) SM010

3. CREATION OF DATA FILES

Data files can exist in one of two forms; as ASCII character files (which must be read and written under format control), or as binary records (read and written without format control).

In general terms data files can, at present, be created in one of the following ways.

3.1. By Editor

The editor program can be used to create data files. The user simply gives the command.

.CREATE filename <cr>

INPUT:

and then types in the required data records. These records are ASCII character strings, each a maximum of 72 characters long, and each terminated by a carriage return.

When the user's program reads this file the records must be read under format control in an identical manner to reading input data cards.

example:

Suppose an input record to the data file was
1234<cr>

Reading this record with the instructions

READ (10, 2) M
2 FORMAT (I4)

would give M the value of 1234

However the same record, read by

```
        READ (10,4) I, G
4        FORMAT (I1, 1X, F2.1)
```

```
would give the result I = 1
G = 3.4
```

3.2 From Cards

If a user has a deck of cards, he can transfer these to a disk file which can then be processed as any other ASCII file.

Facilities will be available in a new version of Batch to enable a user to create a disk file from cards by running a batch job.

As a temporary, interim service until the new Batch is available, the Centre will transfer card decks to disk files for remote terminal users.

3.3 By program

A user's own program can output named data files to the disk. These files can be binary or ASCII and can be kept on the disk for later use by other programs.

4. USE OF DATA FILES IN FORTRAN

To be able to read or write data files with a FORTRAN program, it is necessary to have some mechanism by means of which a specific data file can be assigned to a FORTRAN logical unit number.

For example the user may wish to make the FORTRAN statement

```
READ (10, 3) A, B, C, J
```

read its data records from the user's data file named DATIN. To do this the user must specify that, in this program, the file DATIN is assigned to the logical unit number 10.

In the PDP-10 system these assignments are made by means of the subroutines IFILE and OFILE. File assignments can only be made to the FORTRAN logical unit numbers 10, 11, 12 and 13. The assignment of a file to a logical unit number can be broken by the RELEAS subroutine.

4.1 Subroutine IFILE

The IFILE subroutine assigns a named file to a logical unit number as an input file. This file can be read only on that unit number; it cannot be written.

The calling sequence for IFILE is as follows:

```
CALL IFILE (n, filename)
```

where 'n' is the FORTRAN logical unit number to which the file is being assigned (10, 11, 12 or 13)

and 'filename' is a literal string or a variable containing the name of the file to be assigned in ASCII.

examples:

(i) CALL IFILE (10, 'DATIN')

This assigns the file DATIN to

FORTRAN logical unit number 10.

Each read on unit 10 will read the next sequential record from DATIN.

While this assignment holds unit 10 cannot appear in a WRITE statement.

(ii) DATA INF/'INPUT'/

CALL INFILE (12, INF)

READ (12) ARRAY

Each execution of the READ statement will read the next sequential record from the file INPUT into ARRAY. In this example the file INPUT is a binary file.

4.2 Subroutine OFILE

The OFILE subroutine assigns a named file to a logical unit number as an output file. This file can be written only on that unit number.

The calling sequence for OFILE is similar to that for IFILE, namely

```
CALL OFILE (n, filename)
```

where 'n' is the FORTRAN logical unit number to which the file is being assigned (10, 11, 12 or 13).

and 'filename' is a literal string or a variable containing the filename in ASCII.

examples:

(i) CALL OFILE (10, 'DOUT')

This assigns the file DOUT to the FORTRAN logical unit number 10. Each write on unit 10 will write the next sequential record onto DOUT.

While this assignment holds unit 10 cannot appear in a READ statement.

(ii) DATA NIT, INFO/13, 'DOUT' /
CALL OFILE (NIT, INFO)
WRITE (NIT, 7) ARRAY

Each execution of the WRITE statement will output the contents of ARRAY, under format statement 7, as the next sequential record on the file DOUT.

4.3 Subroutine RELEAS

The RELEAS subroutine breaks the assignment of a named file to a logical unit number. That unit number can then be reassigned by another call to IFILE or OFILE.

The calling sequence for RELEAS is simply

CALL RELEAS (n)

where 'n' is the logical unit number to be released.

example:

```
DATA K, L/10, 11/  
CALL IFILE (K, 'DIN')  
CALL OFILE (L, 'OUTPT')  
DO 2 I = 1, 50  
READ (K, 1) M  
1  FORMAT (I6)  
2  WRITE (L) M  
CALL RELEAS (L)  
CALL IFILE (L, 'OUTPT')  
DO 3 I = 1, 50  
READ (L) M  
3  WRITE (6, 4) M  
4  FORMAT (1X, I6)  
CALL EXIT  
END
```

This sample program reads 50 records from the ASCII file DIN and copies them in binary form to the file OUTPT. The OUTPT file is then read back and printed.

5. SOME DETAILS ON DATA FILE USAGE

There are a number of details to be remembered in the use of data files via the IFILE and OFILE routines.

- (a) Files can only be read sequentially or written sequentially. The first READ or WRITE after the file assignment will process the first record of the file. Subsequent READ and WRITE statements process successive records on a file.
- (b) Files can be in ASCII or binary. They should always be read in the same form as they were written.
- (c) A REWIND statement on a logical unit number can be given, but it will, at present, simply act as a call to RELEASE and break the assignment of named file to logical unit number.
- (d) Files cannot currently be backspaced. The BACKSPACE statement is ignored.
- (e) One file can be used for both input and output by assigning it to two different logical unit numbers.

For example suppose the following assignments had been made.

```
CALL IFILE (10, 'DATA')
CALL OFILE (11, 'DATA')
```

Information could then be read from unit 10, processed, and written to unit 11. As both input and output files have the same name, this operation will cause the file assigned to unit 10 to be superseded by that assigned to unit 11. At the completion of this operation the file named DATA will contain only the information output on unit 11. The original information on the file DATA input on unit 10 will have been irretrievably destroyed. This facility should therefore be used with care.

(f) FORTRAN logical units can be used either as input units or as output units. However a given unit cannot be assigned for both input and output simultaneously.

For example the two statements

```
CALL INFILE (10, 'DATA')
CALL OFILE (10, 'DOUT')
```

are invalid unless separated by a call to
RELEASE.

6. SCRATCH FILES

Output can be done to, (and input from) the logical unit numbers 10, 11, 12 and 13 without first assigning a named file to these units.

In this case the system will automatically assign the following default filenames.

<u>Unit No.</u>	<u>Filename</u>
10	FOR10
11	FOR11
12	FOR12
13	FOR13

If the first operation on a given unit number is a WRITE, a file, named as above, will be written onto the user's disk area. If the first operation on a given unit is a READ an attempt will be made to find the file, named as above, on the user's disk area. If none exists an error message.

?FILENAME FORTR NOT ON DEVICE DSK

will result.

Once files of these names have been produced the user can treat them as ordinary files. Users should be careful not to keep files under these names as they could easily be inadvertently destroyed.

Items (a) (b) (c) and (d) noted in section 5 above apply to these scratch files.

7. AVAILABILITY

These facilities have been implemented and can now be used from remote terminals.

Users are advised that whenever improved file or assignment facilities become available, there may be changes in the operation of the routines outlined above.

A SUGGESTION ON REMOTE TERMINAL I/O WITH FORTRAN

Two problems arise with the use of FORTRAN on the present remote terminal system:

- (a) the entry of data to a program
- (b) the output of a large volume of results on the Teletype

Both these problems may be alleviated somewhat by the use of data files on disk.

Input data files can be created by use of the Editor and read by the user's program.

Similarly for a large volume of output, the output can be written to a disk file, instead of the Teletype and the output file contents inspected with the Editor. Note however that the Editor will truncate lines of more than 80 characters.

PROFESSIONAL DEVELOPMENT SEMINAR

The Queensland Branch of the Australian Computer Society has organized a full-day seminar on *Decision Tables*.

The Seminar has been arranged for Friday 12 February 1971 from 9 a.m. to 8 p.m. A copy of the time table is given below.

The venue will be:

Management House,
Australian Institute of Management,
Corner Boundary/Rosa Street,

AUSTRALIAN COMPUTER SOCIETY INCORPORATED

QUEENSLAND BRANCH

PROFESSIONAL DEVELOPMENT SEMINAR
"DECISION TABLES"

Programme: 9.00 a.m. to 8.00 p.m. FRIDAY, 12 February, 1971.

9.00 - 9.15	Official opening speech	- Mr Ashley Goldsworthy Chairman Q'ld Branch A.C.S.
9.15 - 1.25	Seminar Chairman's Introduction	- Mr Peter Richards, University of Queensland
9.30 -10.00	An Introduction to Decision Tables	- Mr David Abercrombie I.B.M.
10.05 - 10.20	Morning Tea	
10.25 - 11.05	The Use of Decision Tables in Systems Analysis and Design	- Mr Peter Richards, University of Queensland
11.10 - 11.50	Application of Decision Tables Electricity Billing	- Mr Jon Moore, B.C.C.
11.55 - 12.25	Programming with Decision Tables	- Mr Mike McLean University of Queensland
12.30 -1.45	Lunch (Smorgasbord)	
1.50 - 2.10	Current support from I.B.M.	- Mr David Abercrombie, I.B.M.

2.10 - 2.30	Current support from N.C.R.	- Mr Grahame Mackay N.C.R.
2.30 - 2.50	Current support from I.C.L.	- Mr Mike Daniell, I.C.L.
2.55 - 3.40	A Critical Appraisal	- Mr John Barker and Mr Bernard Speight Q.U.F.
3.45 - 4.00	Afternoon Tea	
4.05 - 4.45	Modification of a Preprocessor	- Mr Brian Cruse and Mr Bob Ajax M.R.D.
4.50 - 5.25	Discussion Session in Syndicate Groups	
5.30 - 6.45	Dinner (Buffet)	
6.50 - 7.25	Open Session	
7.30 - 7.50	Observer's Critique	- Mr Alan Coulter, Corrie and Co
7.55 - 8.00	Conference Close	

The cost of the seminar is as follows:

A.C.S. members	\$12.00
non-members	\$15.00

There is a special price of \$5.00 for any students wishing to attend.
However, this does not include the meals indicated in the timetable.

Anybody wishing to attend the seminar should complete the following enrolment form. The closing date for enrolments is 5 February.

ENROLMENT FORM (Enrolments close 5th Feb.1971)

AUSTRALIAN COMPUTER SOCIETY INCORPORATED - QUEENSLAND BRANCH
PROFESSIONAL DEVELOPMENT SEMINAR - "DECISION TABLES"

Friday 12th February 1971 9.00 a.m. - 8.00 p.m.

Venue: Management House (A.I.M.) Cnr. Boundary/Rosa Streets,
SPRINGHILL, Brisbane

NAME:
Surname (please print) Initials Name for lapel badge

ORGANIZATION:

.....

.....

Name and Address

.....

Phone number

ACS Member \$12 - (includes meals and seminar papers)

Non-Members \$15- (includes meals and seminar papers)

ACS Student Members - concessional fee - \$5. (does NOT include meals)

Send remittance with this form to: The Secretary,
A.C.S.,
P.O. Box 1484, G.P.O.,
BRISBANE 4001

